

BAB II

TINJAUAN PUSTAKA

2.1 *Heat exchanger*

Heat exchanger (HE) merupakan instrument yang bisa mengatasi terjadinya gangguan pada *steam supply* pada *boiler* (Nowak, 2013). Secara prinsip, ia dapat mengubah suhu air dalam tangki *heat exchanger* sebagai *variable* yang dikontrol (*controlled variable*). Gangguan yang terjadi dalam *heat exchanger* ini biasanya terdapat pada *controlled variable* (suhu air) dan tekanan *steam supply*.

Heat exchanger merupakan salah satu perangkat penting yang banyak digunakan dalam industri kimia untuk proses pemanasan atau pendinginan cairan (Ismail dkk, 2015). Dalam penelitian ini penulis menggunakan model *heat exchanger* dengan merujuk fungsi alih berikut ini:

$$HE = \frac{1}{30s+1} \quad (2.1)$$

2.2 **Motor Servo**

Motor servo adalah jenis motor DC yang bekerja menggunakan sumber tegangan DC. Di dalam motor servo ini terdapat beberapa bagian seperti motor DC, potensiometer, *gear box*, dan rangkaian kontrol. Bentuk motor servo dapat dilihat pada gambar 2.1.



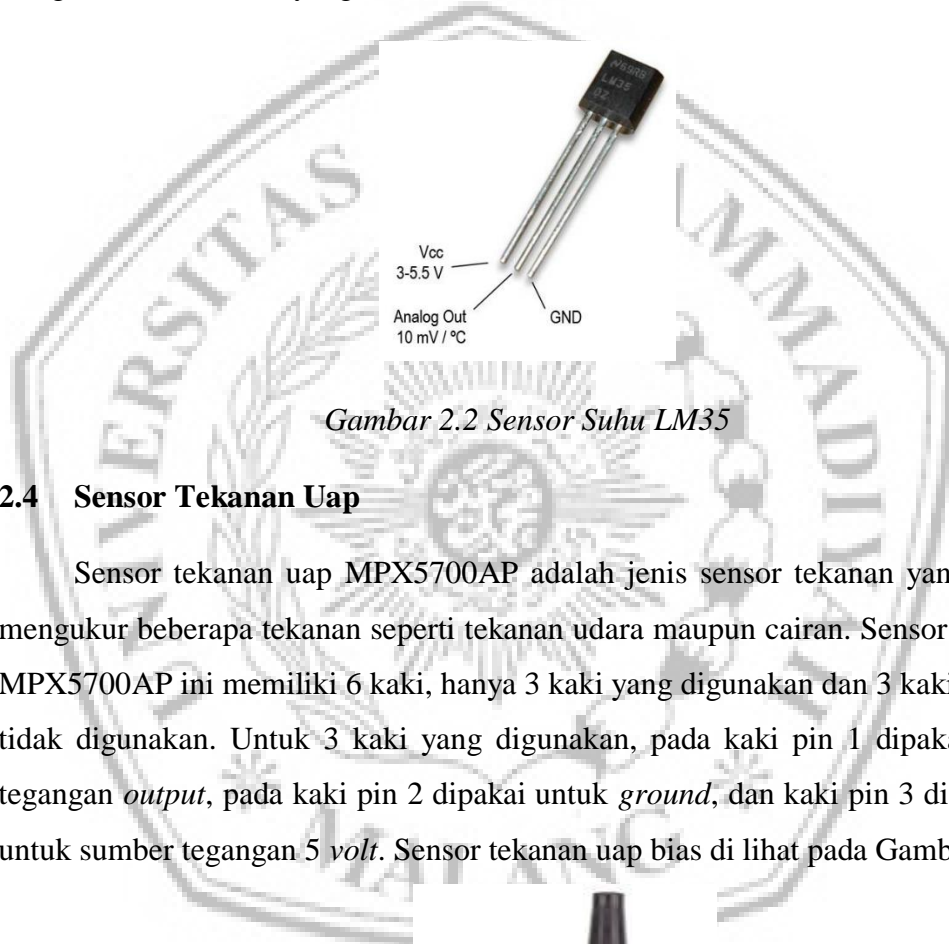
Gambar 2.1 Motor Servo

2.3 Sensor Suhu LM35

Sensor suhu LM35 adalah jenis sensor analog yang memanfaatkan pembacaan suhu kedalam besaran tegangan. *Range* pembacaan sensor ini mulai dari 0°C sampai dengan 150°C. Untuk persamaan dan gambar sensor suhu LM35 ini bisa dilihat pada Persamaan (2.2) serta Gambar 2.2

$$V_{out} = 10mV \times T \quad (2.2)$$

Dengan T adalah suhu yang dideteksi sensor LM35.



Gambar 2.2 Sensor Suhu LM35

2.4 Sensor Tekanan Uap

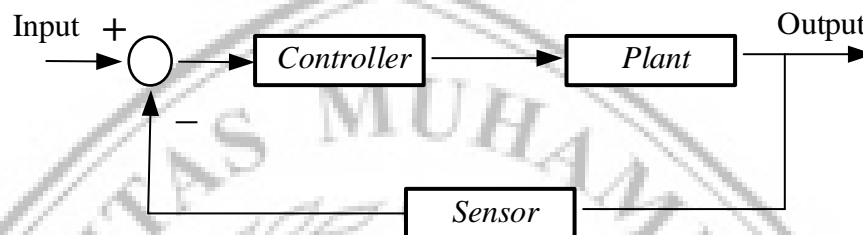
Sensor tekanan uap MPX5700AP adalah jenis sensor tekanan yang dapat mengukur beberapa tekanan seperti tekanan udara maupun cairan. Sensor tekanan MPX5700AP ini memiliki 6 kaki, hanya 3 kaki yang digunakan dan 3 kaki lainnya tidak digunakan. Untuk 3 kaki yang digunakan, pada kaki pin 1 dipakai untuk tegangan *output*, pada kaki pin 2 dipakai untuk *ground*, dan kaki pin 3 digunakan untuk sumber tegangan 5 volt. Sensor tekanan uap bias di lihat pada Gambar 2.3.



Gambar 2.3 Sensor Tekanan Uap MPX5700AP

2.5 Kontrol Feedback

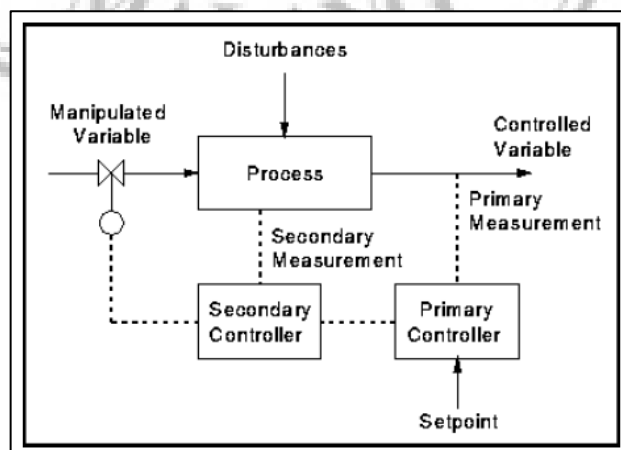
Kontrol *feedback* (Kontrol tertutup) adalah sistem kontrol yang memiliki 3 unsur utama yaitu kontroler, *plant* atau model kontrol, dan sensor sebagai pembacaan sinyal keluaran sistem. Kontrol *feedback* dapat memperkecil kesalahan sistem dengan selisih antara sinyal keluaran dan sinyal masukan, kemudian diumpan balikan ke kontroler supaya dapat memperkecil dan membuat keluaran sistem mendekati masukan (*set point*) yang diinginkan. Untuk blok kontrol *feedback* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Sistem Kontrol feedback (kontrol tertutup)

2.6 Kontrol Cascade

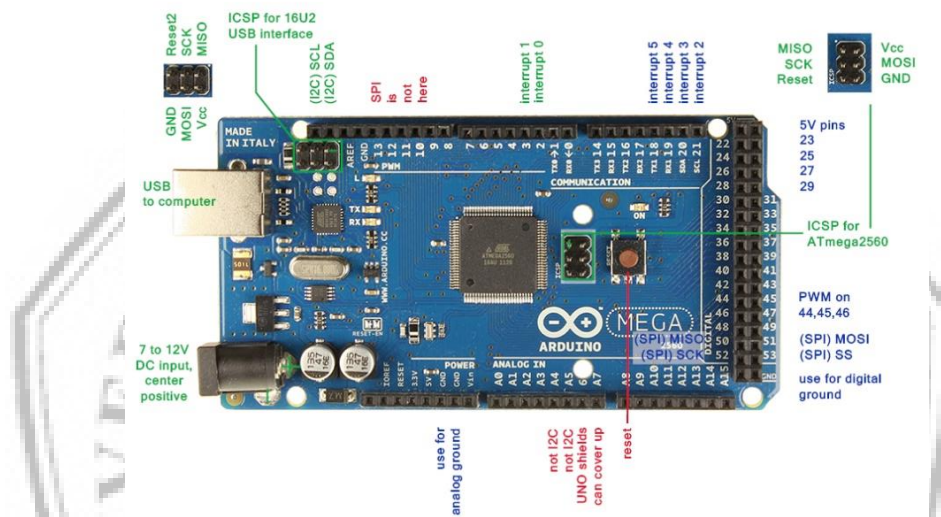
Kontrol *cascade* atau kontrol bertingkat adalah sebuah sistem kontrol yang memiliki 2 kontrol *feedback*. Setiap kontrol *feedback* memiliki kontroler dan sensor masing-masing, yang biasanya disebut dengan kontrol utama (*primary control*) dan kontrol kedua (*secondary control*). Gambar 2.5 Kontrol *cascade* disini dipergunakan untuk sistem yang memiliki tanggapan yang lambat dan sistem yang memiliki gangguan atau perubahan yang mengakibatkan perlambatan sistem proses.



Gambar 2.5 Sistem Kontrol Cascade

2.7 Arduino Mega2560

Arduino Mega2560 adalah papan pengembangan mikrokontroler yang berbasis Arduino dengan menggunakan chip ATmega 2560. Board ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah PWM), 16 pin analog input, 4 pin UART (*serial port hardware*). Arduino Mega 2560 dilengkapi dengan sebuah oscillator 16 Mhz, sebuah port USB, power jack DC, ICSP header, dan tombol reset. Board ini sudah sangat lengkap, sudah memiliki segala sesuatu yang di butuhkan untuk sebuah mikrokontroler.

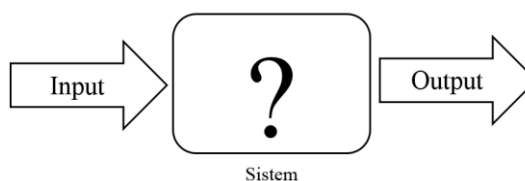


Gambar 2.6 Ardunio Mega2560

Pada Gambar 2.6 menunjukkan secara detail konfigurasi dari PIN-PIN Arduino Mega2560. Arduino pada penelitian ini dipakai sebagai antar muka antara *software* MATLAB dengan *plant* pengendali suhu *heat exchanggr* yang akan dikontrol.

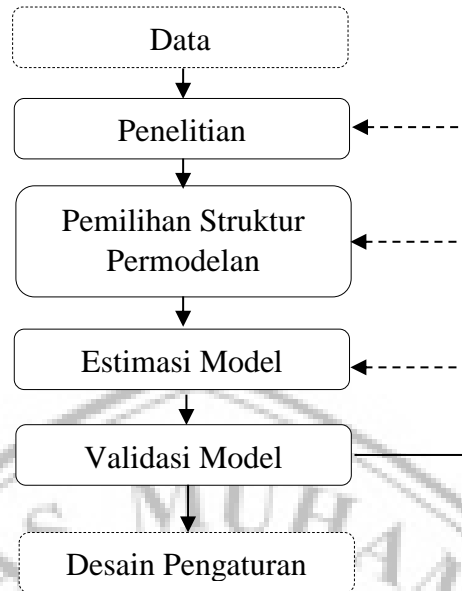
2.8 Identifikasi Sistem

Identifikasi sistem adalah suatu metode untuk menentukan pemodelan matematis dari sebuah sistem dinamis yang parameternya tidak diketahui dengan cara pengolahan data *input* dan *output* dari sistem tersebut pada Gambar 2.7.



Gambar 2.7 Identifikasi Sistem

Secara umum proses identifikasi sistem seperti pada Gambar 2.8 berikut.



Gambar 2.8 Flowchart Identifikasi Sistem

2.8.1 Pengambilan Data

Proses identifikasi sistem memerlukan pengambilan data *input-output*. Data *input-output* diperlukan untuk proses identifikasi dari sebuah sistem, data ini biasanya berupa data dengan kawasan waktu dan frekuensi.

2.8.2 Penelitian

Penelitian dilakukan dalam dua langkah. Pada langkah pertama, penelitian pendahuluan seperti menentukan jenis respon yang digunakan dan langkah yang kedua dilakukan untuk mendapatkan pengetahuan dasar dengan karakteristik yang penting dari sebuah sistem.

2.8.3 Pemilihan Struktur Pemodelan

Pemilihan struktur pemodelan digunakan untuk menentukan pengaturan poin dimana estimasi model dilakukan. Dalam *sistem identification toolbox* struktur pemodelan yang paling umum digunakan adalah model *transfer function*.

$$Y(s) = \frac{Num(s)}{Den(s)} U(s) \quad (2.3)$$

Dimana Y dan U adalah pernyataan *input* dan *output*, dalam penelitian ini jenis pemodelan *transfer function* yang digunakan dengan zeros bernilai 1 dan poles bernilai 2.

2.8.4 Estimasi Model

Estimasi model adalah prosedur mencocokkan model dengan struktur model tertentu. Struktur model disini seperti model linier dan model dengan struktur parameter tertentu misalnya, model fisika, model ARX. Selain model linier parametrik, model linier dapat terdiri juga berasal dari fungsi pembobotan atau fungsi transfer dalam bentuk respon frekuensi. Estimasi bertujuan untuk memperoleh model parameter tertentu dan untuk memperkirakan hubungan *input-output* dalam bentuk *transfer function* (TF).

Nilai ketepatan atau *best fit* diperoleh dari perbandingan data validasi *output* dan respon *output* dari pemodelan dengan rumusan sebagai berikut:

$$fit = \left[1 - \frac{\|y - \hat{y}\|}{\|y - \text{mean}(y)\|} \right] \quad (2.4)$$

Dimana: y = data validasi *output*

\hat{y} = respon *output* dari sistem

2.8.5 Validasi Model

Validasi model setelah memperoleh estimasi pemodelan dengan parameter tertentu, selanjutnya data divalidasi dalam berbagai cara. Dalam hal ini menggunakan model *validity criterion*, *Pole-Zero plot*, *Bode Diagram*, *Residual Analysis* dan simulasi. Sesuai dengan proses identifikasi standar, semua ini digunakan untuk memperoleh hasil pemodelan yang akurat.

2.9 Kontrol Proportional Integral Derivative (PID)

Kontrol *proportional integral derivative* (PID) sendiri merupakan gabungan dari tiga parameter pengaturan yaitu kontrol P(*Proportional*), D(*Derivative*), I(*Integral*). Kontrol *proportional* memberi pengaruh langsung pada *error*. Semakin besar *error*, maka semakin besar sinyal kendali yang dihasilkan oleh kontroler. Adapun persamaan dari kontrol *proportional* seperti pada persamaan (2.5).

$$U_p(t) = K_p e(t) \quad (2.5)$$

Dimana, K_p adalah konstanta *proportional* dan $e(t)$ adalah nilai *error*. Nilai *error* didapat dari nilai *setpoint* dikurangi nilai *output* dari pada sistem. Kontrol

Integral mempunyai karakteristik untuk mengurangi waktu naik (*rise time*), menambah lewatan maksimum (*overshoot*) dan waktu turun, dan menghilangkan kesalahan keadaan tunak (*steady state error*). Persamaan dari kontrol *Integral* seperti pada persamaan (2.6).

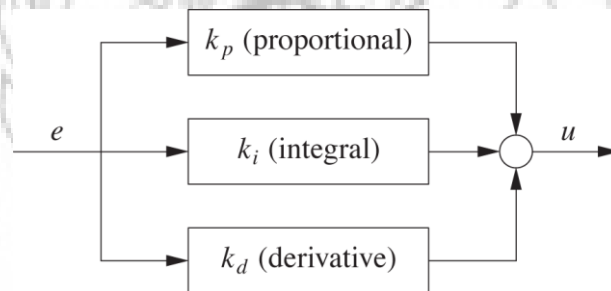
$$U_I(t) = K_i \int_0^t e(t) dt \quad (2.6)$$

Dimana K_i adalah konstanta *integral*. Kontrol *Derivative* akan memberikan efek mengurangi *overshoot* dan meningkatkan stabilitas sistem serta meningkatkan *steady state error*. Persamaan dari kontrol *Derivative* seperti pada persamaan (2.7).

$$U_D(t) = K_d \frac{de(t)}{dt} \quad (2.7)$$

Dimana K_d adalah konstanta *derivative*. Agar kita mendapat hasil *output* dengan waktu naik yang tinggi dan *error* yang kecil, maka dapat digunakan gabungan dari kontrol *Proportional*, *Derivative*, dan *Integral* menjadi kontrol PID. Persamaan dari kontrol PID seperti pada persamaan (2.8).

$$U_{PID}(t) = U_P(t) + U_I(t) + U_D(t) \quad (2.8)$$



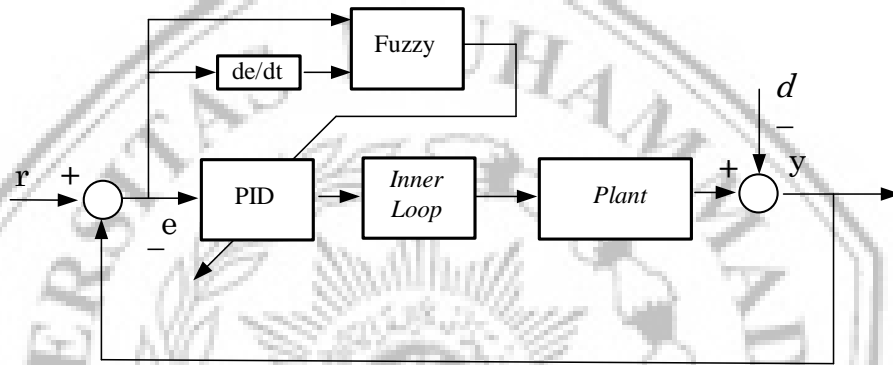
Gambar 2.9 Kendali PID

Tabel 2. 1 Pengaruh PID Pada Waktu Naik, *Overshoot*, Waktu Turun dan Kesalahan Keadaan Tunak

Tanggapan Loop Tertutup	Waktu Naik	<i>Overshoot</i>	Waktu Turun	Kesalahan Keadaan Tunak
Proporsional (K_p)	Menurun	Meningkat	Perubahan Kecil	Menurun
Integral (K_i)	Menurun	Meningkat	Meningkat	Hilang
Derivative (K_d)	Perubahan Kecil	Menurun	Menurun	Perubahan Kecil

2.10 Fuzzy-PID

Fuzzy-PID adalah suatu teknik kontrol kombinasi antara kontrol *fuzzy* dan kontrol PID. Pada kontrol *fuzzy*-PID ini parameter K_p , K_i dan K_d pada kontrol PID diatur menggunakan kontrol *fuzzy*. Dengan demikian parameter dari kontrol PID dapat beradaptasi dengan kondisi *plant* yang dapat berubah-ubah. Pengaturan parameter PID dengan *fuzzy* ini diharapkan dapat lebih akurat dalam memberi sinyal kontrol kepada *plant* agar *plant* dapat mencapai titik *setpoint* dan titik stabil dengan lebih baik. Struktur dari *fuzzy*-PID dapat dilihat pada Gambar 2.10.



Gambar 2.10 Blok kontrol Fuzzy-PID

$$K'_p = \frac{K_p - K_{p \min}}{K_{p \max} - K_{p \min}} \quad (2.9)$$

$$K'_i = \frac{K_i - K_{i \min}}{K_{i \max} - K_{i \min}} \quad (2.10)$$

$$K'_d = \frac{K_d - K_{d \min}}{K_{d \max} - K_{d \min}} \quad (2.11)$$

Persamaan (2.9), (2.10), dan (2.11) menunjukkan bagaimana pengaturan pembaharuan parameter kontrol PID pada kontrol *fuzzy*-PID. Dengan pembaharuan parameter tersebut diharapkan dapat membuat sistem kontrol *fuzzy*-PID lebih kokoh dan handal dalam mengontrol suatu sistem.

2.11 Tinjauan Penelitian Sebelumnya

Sebagai referensi pada penelitian ini digunakan sumber-sumber referensi baik dari jurnal dan buku teks. Penelitian tentang pengaturan pada *heat exchanger* sudah banyak dilakukan dengan berbagai teknik kontrol yang digunakan. Penelitian yang

dilakukan oleh (Zhang, 2012) yang membahas tentang *superheated steam* dengan metode kontrol *neuro-PID*. Penelitian lain yang membahas tentang pengaturan *heat exchanger* juga di angkat oleh (Kumar, 2013) menggunakan teknik kontrol *cascade PID*.

Adapun teknik kontrol pada penelitian ini seperti pada rumusan masalah yang terdapat pada bab sebelumnya. Penggunaan teknik kontrol tersebut tak lepas dari literatur dari penelitian-penelitian sebelumnya yang telah dilakukan. Literatur yang menjadi acuan dalam penelitian yang saya lakukan ini antara lain, penelitian yang telah dilakukan peneliti sebelumnya yaitu (Hasanjani dkk, 2014) dan (Zulfatman dan Rahmat, 2009) dengan teknik kontrol *self-tunning fuzzy PID*, (Cai dan Wang, 2013) yang mengangkat teknik kontrol *hybrid fuzzy-PI*.

